

**LISTING OF THE CLAIMS:**

1. (Currently Amended): A method for providing a meta-data programming language level interface, the method comprising:

receiving an object name from a client program via a meta-data retrieval API, wherein said object name corresponds to a runtime ~~an~~ object instance located in a runtime environment that includes one or more methods;

requesting meta-data associated with said runtime object instance from said runtime environment;

receiving said meta-data for each said method included in said runtime object instance; and

transmitting said meta-data for each said method to said client program via said meta-data retrieval API.

2. (Original): The method of claim 1 wherein said client program is written in Java.

3. (Original): The method of claim 2 further comprising:

creating an internal data structure of the input class byte array in response to said requesting; and

extracting a field and method data from said input class byte array to create said meta- data.

4. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java meta- data class.

5. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java interface meta-data class.

6. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java method meta-data class.
7. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java field meta-data class.
8. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java meta-data type.
9. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java interface meta-data type.
10. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java parameter meta-data type.
11. (Original): The method of claim 2 wherein said meta-data retrieval API includes a Java exceptions meta-data type.
12. (Original): The method of claim 1 wherein said client program is written in C++.
13. (Original): The method of claim 12 wherein said meta-data retrieval API includes a C++ meta-data class.
14. (Original): The method of claim 12 wherein said meta-data retrieval API includes a C++ meta-data type.
15. (Original): The method of claim 1 further comprising retrieving said meta-data.
16. (Original): The method of claim 1 further comprising retrieving said meta-data in a purely abstract manner.

17. (Original): The method of claim 1 wherein said meta-data describes a parameter.

18. (Original): The method of claim 1 wherein said meta-data describes an exception.

19. (Original): The method of claim 1 wherein said meta-data describes a return type.

20. (Original): The method of claim 1 wherein said meta-data includes source code.

21. (Currently Amended): A computer program product for providing a meta-data programming language level interface, the computer program product comprising:

a storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

receiving an object name from a client program via a meta-data retrieval API, wherein said object name corresponds to an a runtime object instance located in a runtime environment that includes one or more methods;

requesting meta-data associated with said runtime object instance from said runtime environment;

receiving said meta-data for each said method included in said runtime object instance; and transmitting said meta-data for each said method to said client program via said meta-data retrieval API.

22. (Currently Amended) A system for providing a meta-data programming language level interface, the system comprising:

a processor in communication with a client system and a runtime system, the processor including instructions to implement a method including:

receiving an object name from a client program via a meta-data retrieval API, wherein said object name corresponds to ~~an~~ a runtime object instance located in a runtime environment that includes one or more methods, said client program is located on the client system, and said runtime environment is located on the runtime system;

requesting meta-data associated with said runtime object instance from said runtime environment;

receiving said meta-data for each said method included in said runtime object instance from said runtime environment; and

transmitting said meta-data for each said method to said client program via said meta-data retrieval API.